

RECEIVED
CENTRAL FAX CENTER

MAR 13 2007

Application No.: 10/630,913

Docket No.: 200313704-1 (1509-437)

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A method of ~~optimising~~ optimizing the performance of an Interpreter-based runtime system, said runtime system including a virtual machine, the virtual machine adapted to run an application in the context of the runtime environment, the method comprising augmenting the bytecode set of the virtual machine with application-specific opcodes by reference to said application, thereby constituting an application domain-specific virtual machine.
2. (Currently Amended) ~~[[A]]The~~ method as claimed in claim 1 wherein the virtual machine is a Java Virtual Machine.
3. (Currently Amended) ~~[[A]]The~~ method as claimed in claim 1 wherein a new application domain-specific virtual machine is generated for different categories of applications.
4. (Currently Amended) ~~[[A]]The~~ method as claimed in claim 1 wherein the dynamic and/or static ~~behaviour~~ behavior of the application is used to create new opcode for the application domain-specific virtual machine.
5. (Currently Amended) ~~[[A]]The~~ method as claimed in claim 1 wherein the virtual machine is optimized based on the hierarchy of the architecture for which the runtime environment is adapted and/or the semantics of the application which is to be run on it.
6. (Currently Amended) ~~[[A]]The~~ method as claimed in claim 1 wherein the virtual machine is optimized based on a late-binding or dynamic loading model and runtime constant manifestation.

Application No.: 10/630,913**Docket No.: 200313704-1 (1509-437)**

7. (Currently Amended) ~~[[A]]The~~ method as claimed in claim 1 wherein semantically enriched code is statically embedded ~~in the application to~~ enable it to run fast~~[[er]]~~ on the application domain-specific virtual machine which is newly generated in accordance with claim 1.
8. (Currently Amended) ~~[[A]]The~~ method as claimed in claim 1 wherein semantically enriched code is dynamically embedded ~~in the application to~~ enable it to run fast~~[[er]]~~ on the application domain-specific virtual machine, said virtual machine newly generated in accordance with claim 1.
9. (Currently Amended) ~~[[A]]The~~ method as claimed in claim 7 wherein the semantically enriched code is determined by performing a quantitative trade-off between time and space.
10. (Currently Amended) ~~[[A]]The~~ method as claimed in claim 7 wherein the semantically enriched code is determined based on the dynamic and/or static ~~behaviour~~ behavior of the application.
11. (Currently Amended) ~~[[A]]The~~ method of generating an embedded virtual machine for a specific domain of ~~applications based on an application, comprising the step of~~ embedding semantically enriched code in ~~the an~~ interpreter loop of the virtual machine.
12. (Currently Amended) ~~[[A]]The~~ method as claimed in claim 11 wherein the semantically enriched code embedding step is performed dynamically on newly loaded portions of the application in dynamic languages.
13. (Currently Amended) ~~[[A]]The~~ method as claimed in claim 12 wherein the interpreter is dynamically enhanced.
14. (Currently Amended) ~~[[A]]The~~ method as claimed in claim 11 wherein secondary codes are used to accommodate the interpretation of new semantically enriched codes.

Application No.: 10/630,913**Docket No.: 200313704-1 (1509-437)**

15. (Currently Amended) ~~[[A]]~~The method as claimed in claim 14 wherein the encoding of the new semantically enriched codes of the instruction set of the virtual machine is performed for efficient decoding of ~~the most~~ frequently executed codes.

16. (Currently Amended) ~~[[A]]~~The method as claimed in claim ~~claim~~ 14 wherein if~~[[.]]~~ a particular code is used ~~very~~ frequently, it is made into a single byte code and the rest of the semantically enriched codes are accommodated by secondary codes.

17. (Currently Amended) A method of ~~optimising~~optimizing the performance of an application~~[[s]]~~ running on an interpreter-based runtime system, the method comprising augmenting the bytecode set of the interpreter with application-specific opcodes by reference to said application, thereby constituting an application domain-specific virtual machine.

18. (Cancelled).

19. (Cancelled).

20. (Currently Amended) A computer program, recorded on a computer-readable medium, ~~adapted~~ to perform the method as claimed in claim 17.